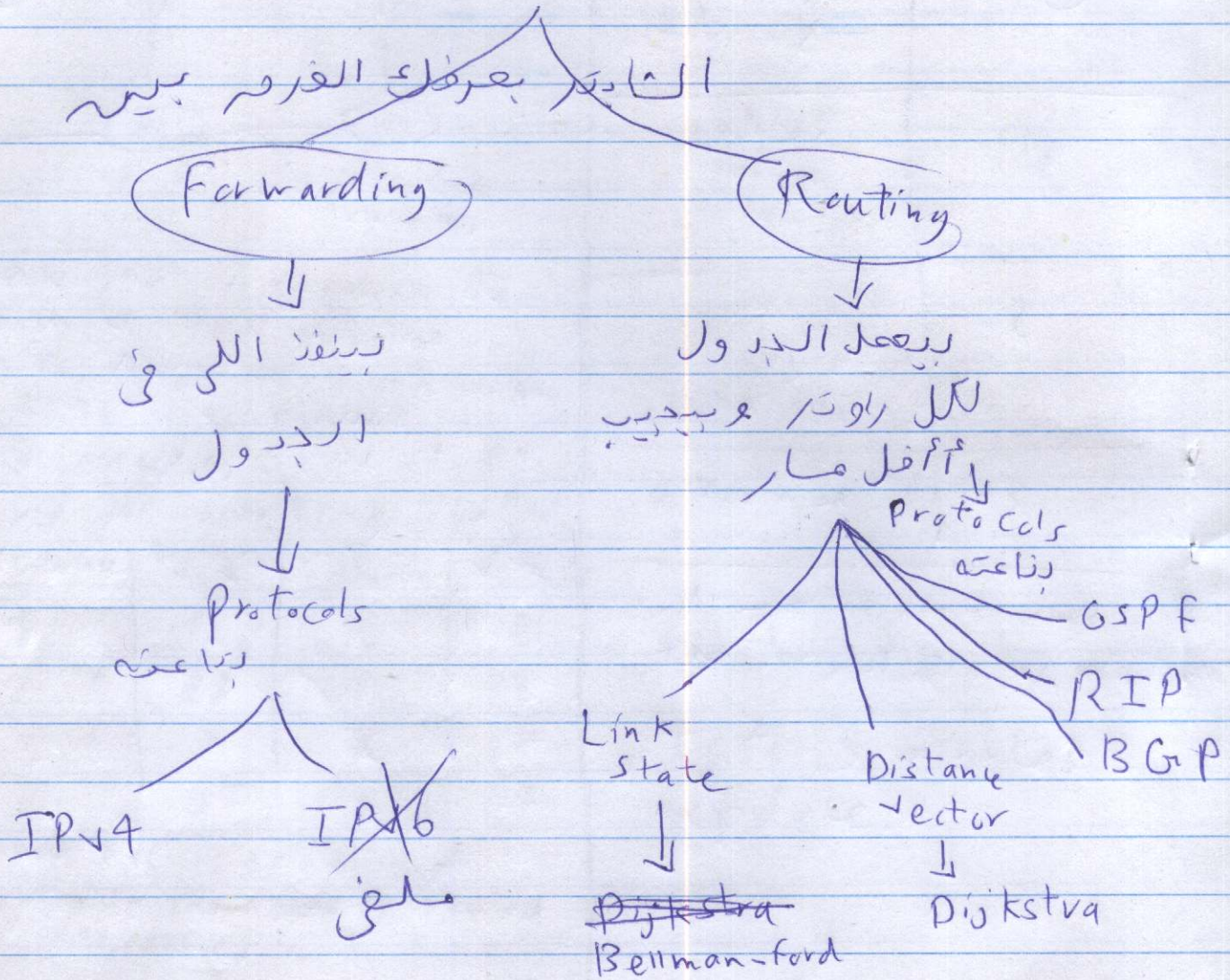
(6)

Chapter 4

ذكر بطة الـ Chapter

الشابتر بعرفك الفرق بينه

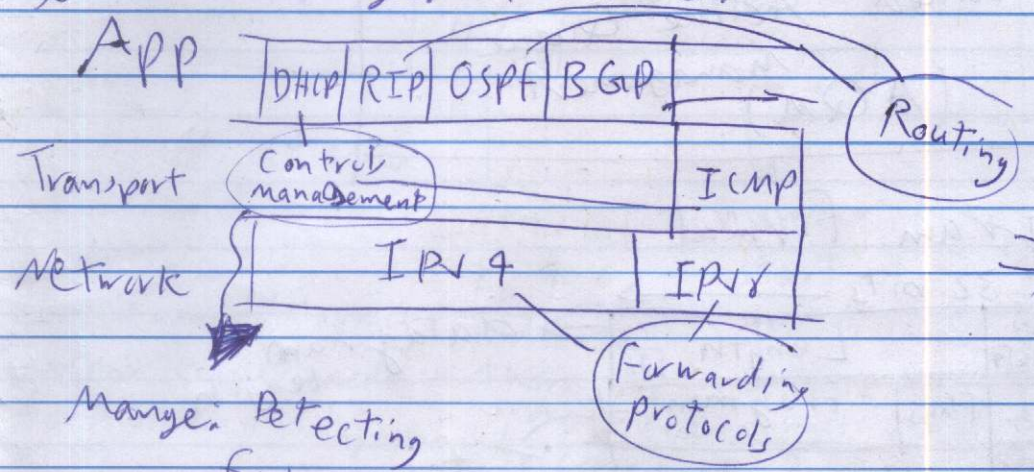


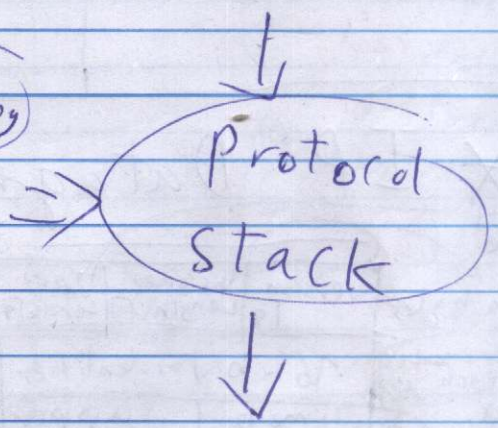
لو لأي هم و احنوى اركبر
مع حبة نظري

Networks Chapter 4 **

- Routers do not run app., transport layers
- Routing: Preparing Forwarding table "who is connected to what" "Determining the path"
- Forwarding: Look up the table and find where to send (Address prefix lookup) (using the table)

* Network layer Protocols:





Manage: Detecting faults, all is alright (optional → when all is right)

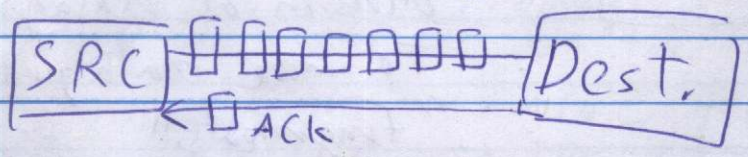
- Control: Required part added by the network "header" or "additional Control Packets" "Nt data"

* Forwarding: We have table like this "Forwarding table"

Prefix	Next Router	Interface
126.23.45.67/32	125.200.1.1	1
128.272.15.0/24	125.200.1.2	2
128.272/16	125.200.1.1	1

"Based on Longest Prefix Match: which means longest parts which matches"

* Ideal Buffering:



→ Flow Control Buffering = RTT * C
"one user" (C: Transmission Rate)

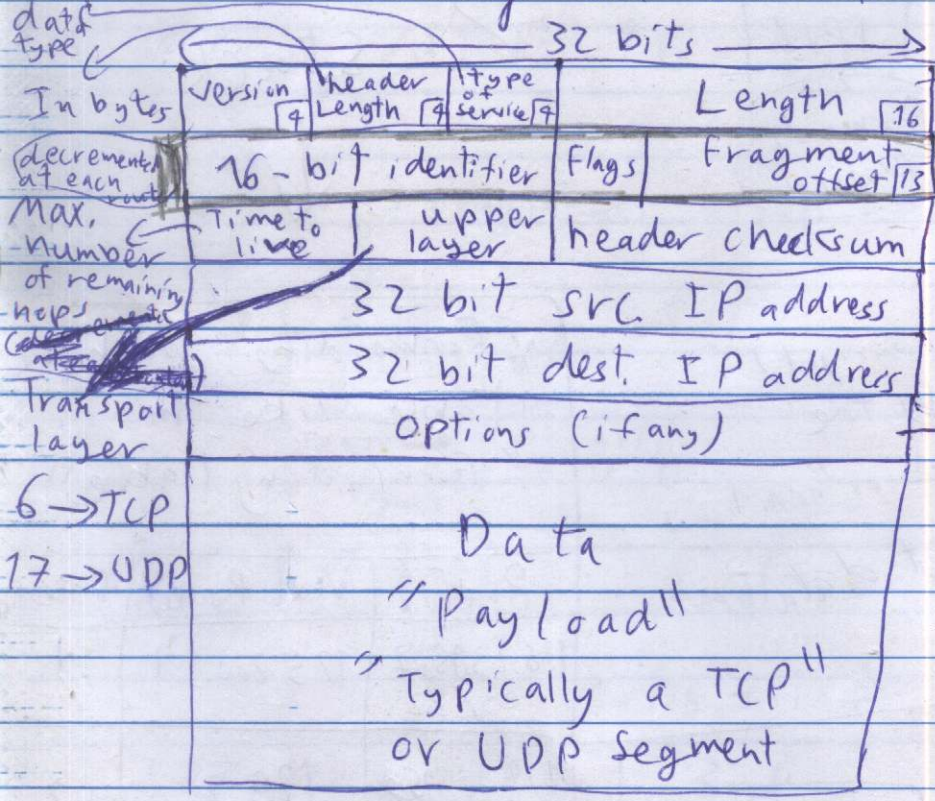
→ Buffer = RTT * C / √(No. of TCP flows)
"many users"

* Packet Dropping Policies:

(*NOL)

Drop Tail	Random Early Drop (RED)
Drop the arriving packet when queue is full.	Drop arriving packets even before queue is full according to probability: Drop probability / Average Q / "Queue size"
	"No choice" if queue is full
	Called "Active Queue Management" (AQM)

* IP Datagram Format



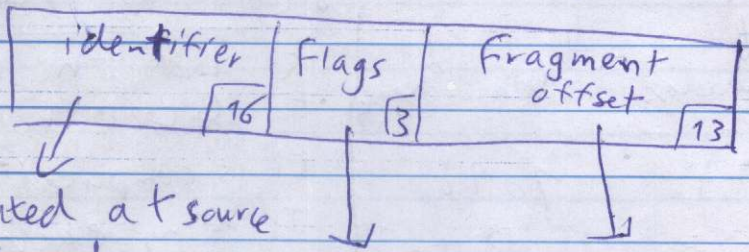
"IPv4"

* Fragmentation: It is division of datagram to more datagrams "fragments"

Note: → small numbers like [version 4] indicates how many bits for each field (not required to be written in exam)

* IP Framentation Format:

- Maximum Transmission Unit (MTU): it is the number of maximum amount of data can be carried by a link layer datagram. (Differs between routers and protocols)



- Created at source host and incremented at each datagram
- when router needs to divide a datagram it leaves it as it is for all new fragments
- At destination, identifiers are examined to determine where are the fragments of a complete datagram

- It is set to 1 to indicate there is more fragments to come next
- then it is set to 0 at the last fragment in order to make sure that all fragments reached the dest.

- number of start of the fragment
- It is a multiple of 8 bytes

* example: we have: • Datagram length = 4000 Bytes
• MTU = 1500 Bytes
• ID = 777

$\frac{4000-20}{1500} = 2.67$ (20 = the header length) → So, we divide the datagram in to 3 fragments. Each has

- **Version Number:** IP protocol version to enable the router from knowing how to interpret the datagram = 4
- **Header length:** 5 without options = typically 20 Bytes
- **Type of Service (TOS):** to distingaish datagrams based on
  - low delay
  - high throughput
  - Reliability
- **Datagram length:** Length of header + data → max. of 2^{16} → datagrams are rarely larger than 1500 Bytes.
- **Time to live (TTL):** Number of hops to reach dest, and decremented at each router and at TTL = 0 the datagram must be dropped to prevent it from circulating forever.

So, we have: For header ↓

4000 Bytes − 20 Bytes

= 3980 Bytes

~~3 fragments~~

1480 + 20 | 1480 + 20 | 1020 + 20

↓ data ↓ header ↓ data ↓ header ↓ data ↓ header

1500 Bytes | 1500 Bytes | 1040 Bytes

Because MTU = 1500 Bytes.

So, we have 3 fragments which have:

ID	Flag	offset	
777	1	0	1st

1 → there is a next fragment

0 → data starts at byte 0

777	1	185	2nd

185 → data starts at byte: 8 * 185 = 1480

777	0	370	3rd

0 → there are no fragments after this fragment

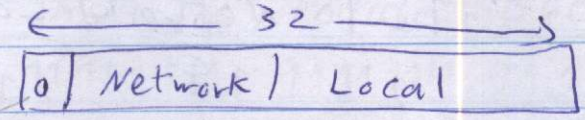
370 → data starts at byte: 8 * 370 = 2960

because, 1st has 1480
2nd has 1480
∴ 3rd starts at

- **Upper layer Protocol:** indicates the specific transport layer protocol for this datagram.
  - 6 → TCP
  - 17 → UDP
- **Header checksum:** to aid the router to detect bit errors. It is computed by treating each 2 header bytes as a number and summing them using 1's complement arithmetic. Datagrams with errors are discarded by the routers. It is recomputed and stored again at each router.
- **Options:** used rarely. It needs more processing so, this field is ignored at IPv6.
- **Data (Payload):** Contains the transport layer segment (TCP or UDP), can carry other types of data like ICMP messages

~~IP Addrers classes~~

IP Addrers classes:



8 bits 8 bits 8 bits 8 bits

- Class A: $\Rightarrow X.X.X.X/8$

$\Rightarrow$ subnet Mask:
255.0.0.0

ex: 10.0.0.1

in binary:

00001010.00000000.00000000.00000001

network part and cannot be changed

Local Part and you can change it as you need to addrers up to 2^{24} devices.

- meaning of subnet mask:
  255.0.0.0
  in binary:
  11111111.0.0.0

by anding the address with the subnet mask we get that $\Rightarrow$ 10.0.0.0 so this part 10.0.0.0 cannot be changed

- Also can be written as 10.0.0.1/8 which means 10.0.0.1/8 the 10 part can not be change which means:
  - IP Addrogs: 10.0.0.1 with subnet mask:
    255.0.0.0

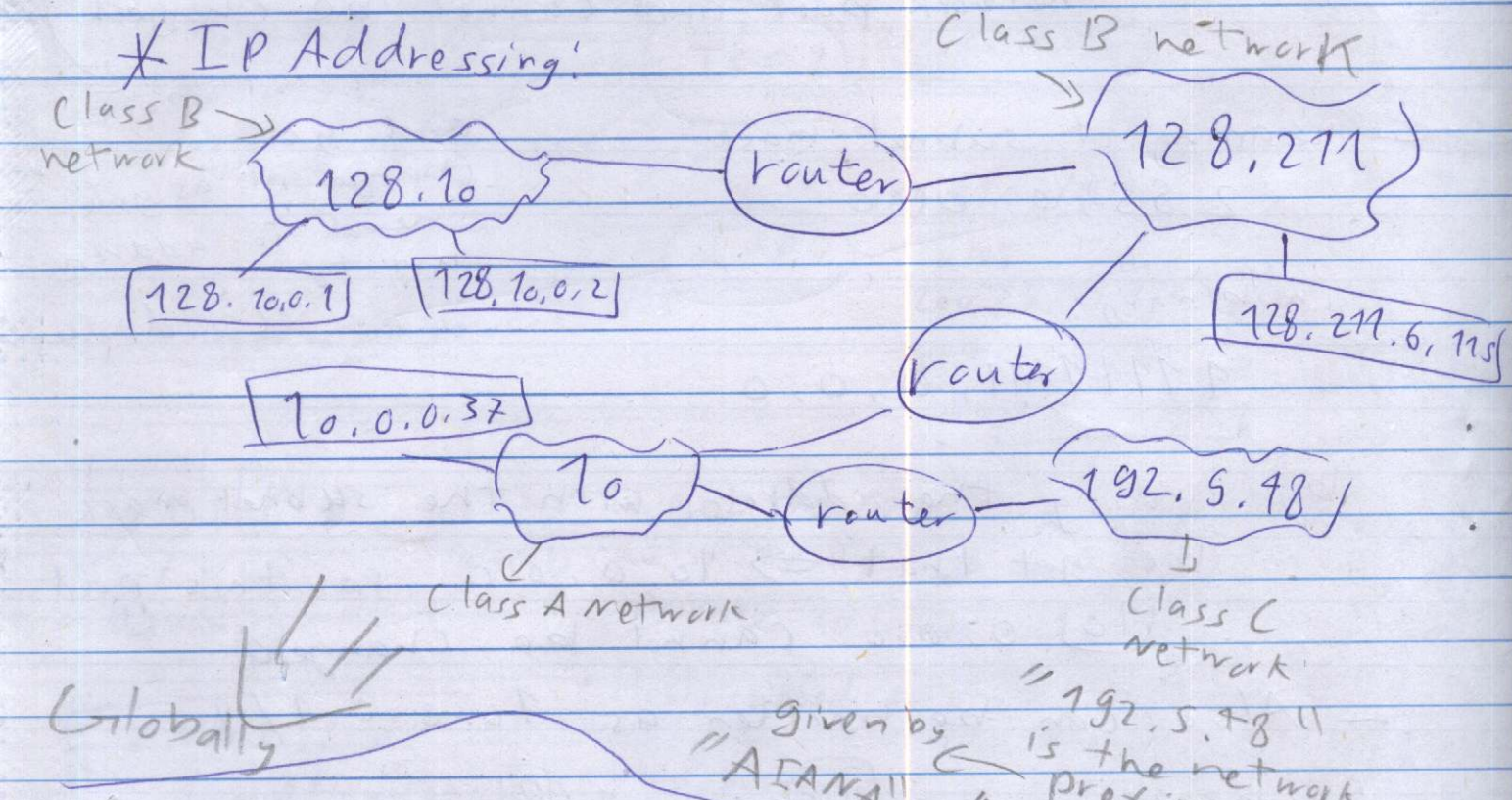
- Class B: | 10 | network | Local | ⇒ X.X.X.X/16
  2 bits, 14 bits, 16 bits ⇒ 255.255.0.0
  up to 2^{16} devices.
- Class C: | 110 | network | Local | ⇒ X.X.X.X/24
  3 bits, 21 bits, 8 bits ⇒ 255.255.255.0
  up to 2^8 devices.

ex: 192.168.1.7

↓ in binary

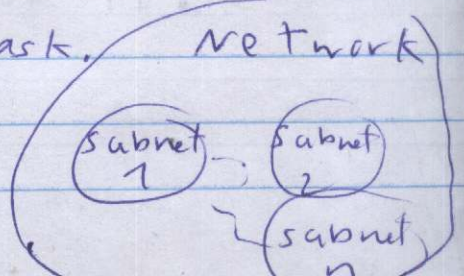
→ 11000000.10101000.00000001.00000111

* IP Addressing:



* Subnetting: All hosts on a subnetwork have the same prefix.
→ prefix is indicated by a subnet Mask.

ex: First 23 bits → subnet.



01

- Address: 10010100. 10101000. 00001000. 11110001
- Mask: 11111111. 11111111. 11111110. 00000000
- ANDing: 10010100. 10101000. 00001000. 00000000

→ using subnetting mask notation is better in order to specify where zeros and ones but slash notation (X.X.X.X/[]) only can help us to specify the number of first bits which form the prefix.

~~→ Multiple subnets require multiple routers~~

*CIDR: Classless Inter Domain Routing

- subnet portion of address of arbitrary length using the slash notation: X.X.X.X/[] → can be any number

ex: 200.23.16.0/23

11001000.00010111.00010000.00000000

subnet part | host part (local)

→ Note: All 1's in host part are subnet broadcast address

⇒ broadcast: 11001000.00010111.00010001.11111111

200.23.17.255

⇒ also network address is All 0's in host part.

⇒ the other possibilities are for devices = $2^9 - 2$ devices

* Route Aggregation "Address Aggregation":

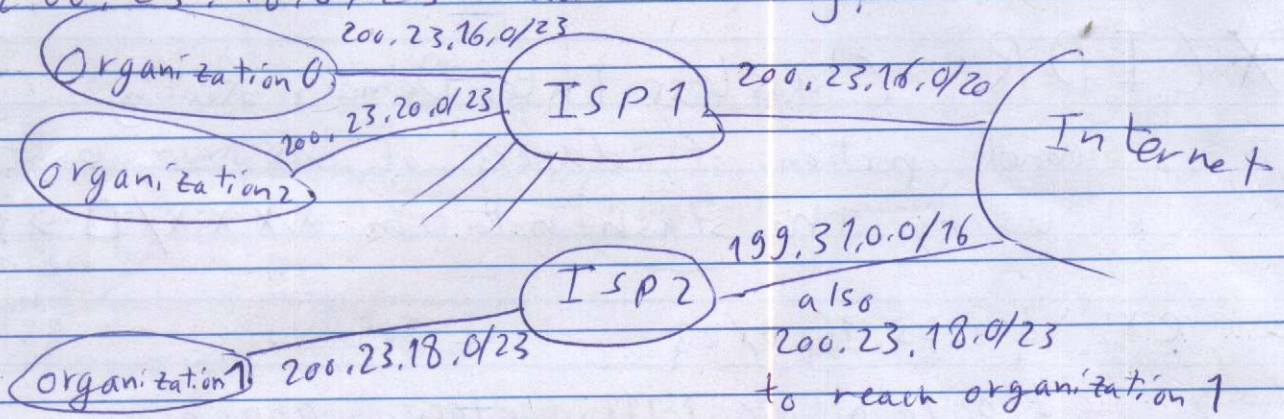
we have: the internet, ISP1, ISP2, Organizations which need their own networks

→ ISP 1: can address in the range: 200.23.16.0/20

→ ISP 2: can address in the range: 199.31.0.0/16

(Prefixes of their organizations' networks are combined within these two shortest (as possible) prefixes)

But, if we have an organization "organization 1" which has its network as: 200.23.18.0/23 which has to be under ISP 1 but, unfortunately it is under ISP2 so, it can be addressed through ISP2 but by its longest prefix: 200.23.18.0/23 as following:

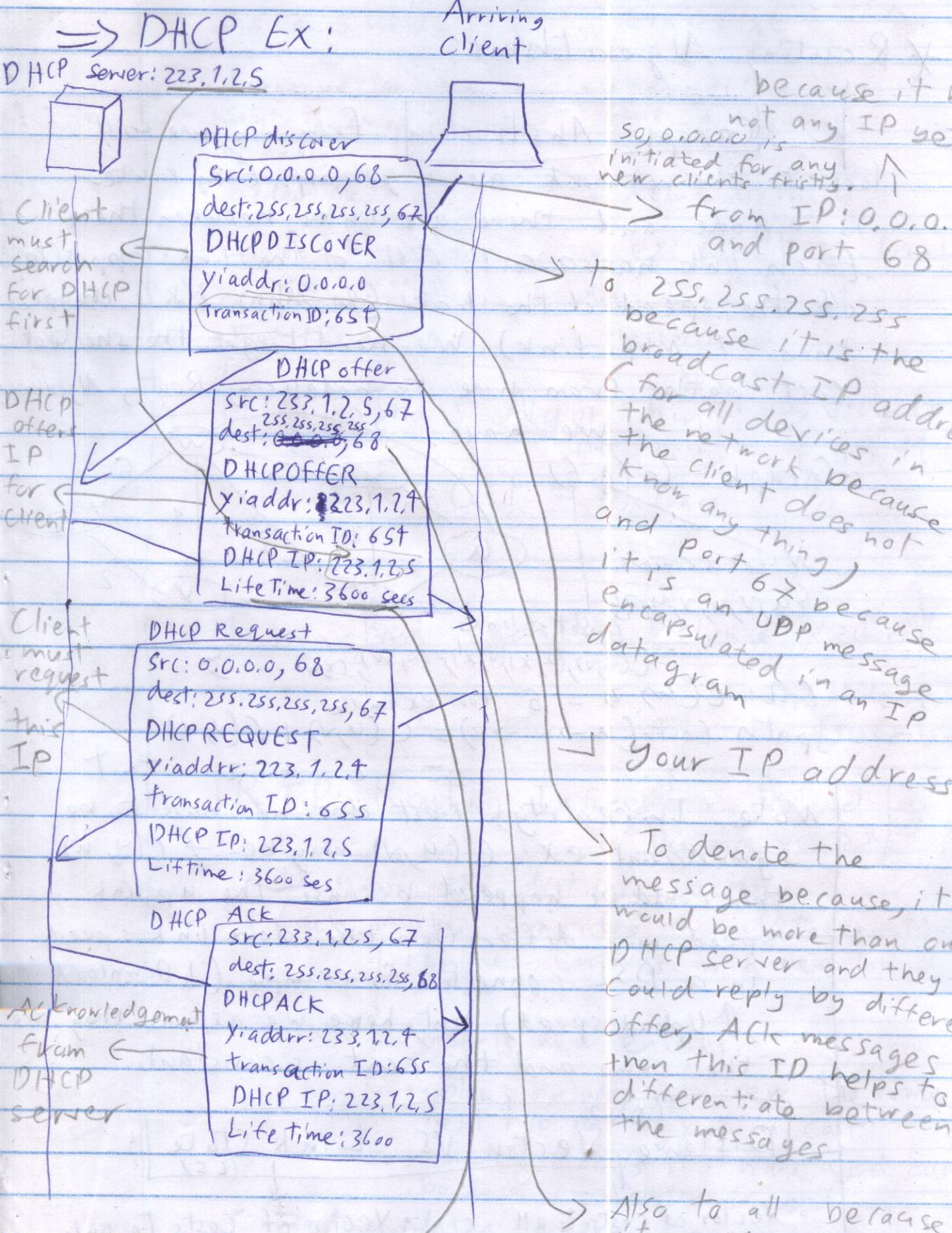


~~DHCP~~ (Dynamic Host Control Protocol): used to get the temporary address you need automatically from a server, used for private addresse and public addresses

- Hosts broadcast: Is there a DHCP server here?
- DHCP servers respond.
- DHCP server keeps the list of assigned addresses (using MAC addresses). It prefers to give you the same address you had last time unless somebody else has taken it
- Lease time: amount of time for the assigned IP address to be valid "several hours or days"

- Note:
  - Multicast: for a group
  - Broadcast: for all

⇒ DHCP Ex:



because it has not any IP yet so, 0.0.0.0 is initiated for any new clients tasks.

→ from IP: 0.0.0.0 and port 68

→ to 255.255.255.255 because it is the broadcast IP address (for all devices in the network because the client does not know any thing) and port 67 because it is an UDP message encapsulated in an IP datagram

→ your IP address

→ To denote the message because, it would be more than one DHCP server and they could reply by different offer, ACK messages then this ID helps to differentiate between the messages

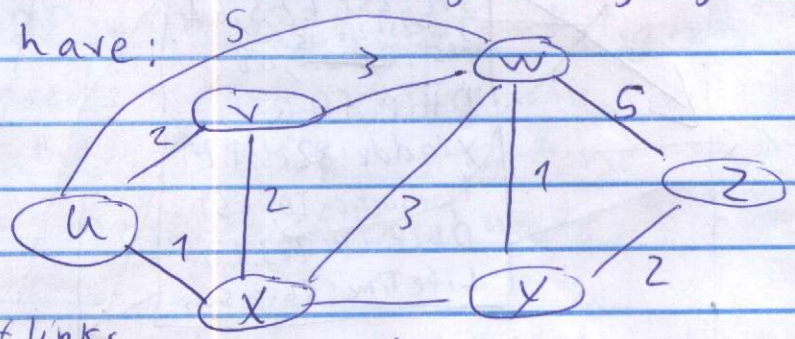
→ Also to all because it would be more than DHCP server.

The lease time for the offered IP

(10)

* Routing Algorithms:

Graph Abstraction: From now, we will look to the network as a graph. Each router is a node and there are links between them. Each link has a cost. Cost of the link: opposite of the speed of the link (10 Mbps link is cheaper than 1 Mbps link). We need to get the shortest cost path from node to node using Routing Algorithms; we have:



- Graph = G(N, E)
  - N: set of routers {u, v, w, x, y, z}
  - E: Set of links {(u,v), (u,x), (v,x), (v,w), (x,w) ---}
- Cost: c(w, z) = 5 for example
- path cost (u → v → x) = c(u,v) + c(v,x) = 2 + 2 = 4

Note: In reality, cost does not have to be symmetrical ex: c(u,v) may be ≠ c(v,w) this could be happened because the up link speed may differ from the down link speed in a DSL connection for example (↓ Download speed, ↑ Upload speed) but, here, we assume they are the same and the cost is constant.

Distance Vector (DV) VS. Link State (LS):

- Vector of costs to all nodes: ex u: {u:0, v:2, w:5, x:1, y:2, ...}
- Vectors of costs to all ~~are~~ sent to only neighbors.

- Vector of costs to only neighbors: ex u: {v:2, w:5, x:1}
- vector of costs to neighbors is sent to all the nodes

11

Distance vector	Link state
• Large vectors are sent to small number of nodes	• small vectors are sent to big number of nodes
• older method	• Newer method
• ex: RIP — protocol (routing Information) "Tell the neighbors about the all"	• ex: OSPF — first (open shortest Path) "Tell the all about the neighbors"

Dijkstra's Algorithm "Link State": ⇒ video 1 [58:14]

Note: Cost ≡ Distance

Bellman-Ford Algorithm "Distance vector":

⇒ video 2 [1:06 – 4:23]

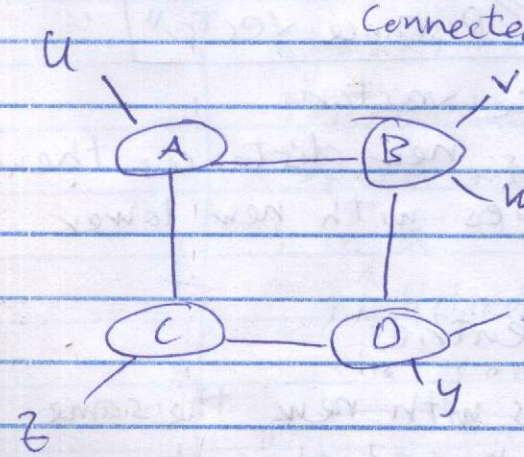
RIP: Routing Information Protocol "Distance vector":

- It uses the distance vector
- Each router computes new distances then:
  - replaces entries with new lower hop counts
  - inserts new entries
  - replaces entries with new the same next hop but higher in cost
  - removes entries that have aged out because each entry is aged
- Send updates every 30 seconds (advertisement) from a specific router
- Note: if no advertisement heard after 180 seconds then this neighbor is considered as dead.

- Disadvantages:
  - Maximum network diameter = 15 hops
  - cost is measured in hops (only hop concept is applied here) → shortest routes may not be the fastest routes (small number of hops but with low speed)
  - Entire tables are broadcast every 30 seconds which leads to having ~~anexha~~ a lot of bandwidth used
  - Uses UDP with 576-byte datagrams so, it needs multiple datagrams to send tables ex: 300-entry table (table which has 300 entries) needs 12 datagrams
  - An error in one routing table is propagated to all routers
  - Slow convergence

RIP ex: we have sabnets: (u, v, w, x, y, z)

Connected in this way:



Note: a hop is a connection between 2 links "network" (number of routers)

So, the table of A could be:

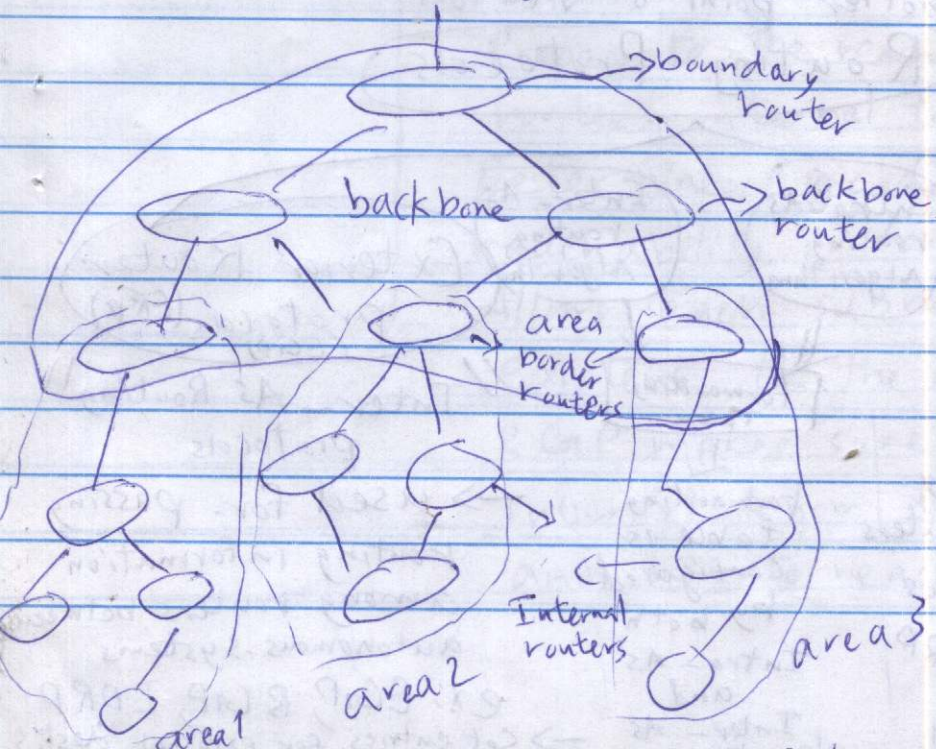
Dest. Subnet	Next Router	hops to Dest.
u	—	1
v	B	2
w	B	2
x	B or C	3
y	B or C	3
z	C	2

- Note: RIP is included only in BSD-UNIX distributions

OSPF: Open Shortest Path First "Link State"

- IS-IS protocol is similar to OSPF

- Uses true metrics (not just hop count) to calculate cost
- Uses subnet Masks
- Allows load balancing across equal-cost paths "multiple same cost-paths"
- Supports type of service (ToS)
- Allows external routes (routes learnt from other autonomous systems)
- Authenticates route exchanges "messages" which leads to security
- Quick convergence
- Direct support for Multicast
- It uses flooding of link-state information and Dijkstra's least-path algorithm
- OSPF advertisement carries one entry per neighbor
- Integrated uni- and multi- cast support
- Hierarchical OSPF: "in large domains"



local area → two level hierarchy
backbone

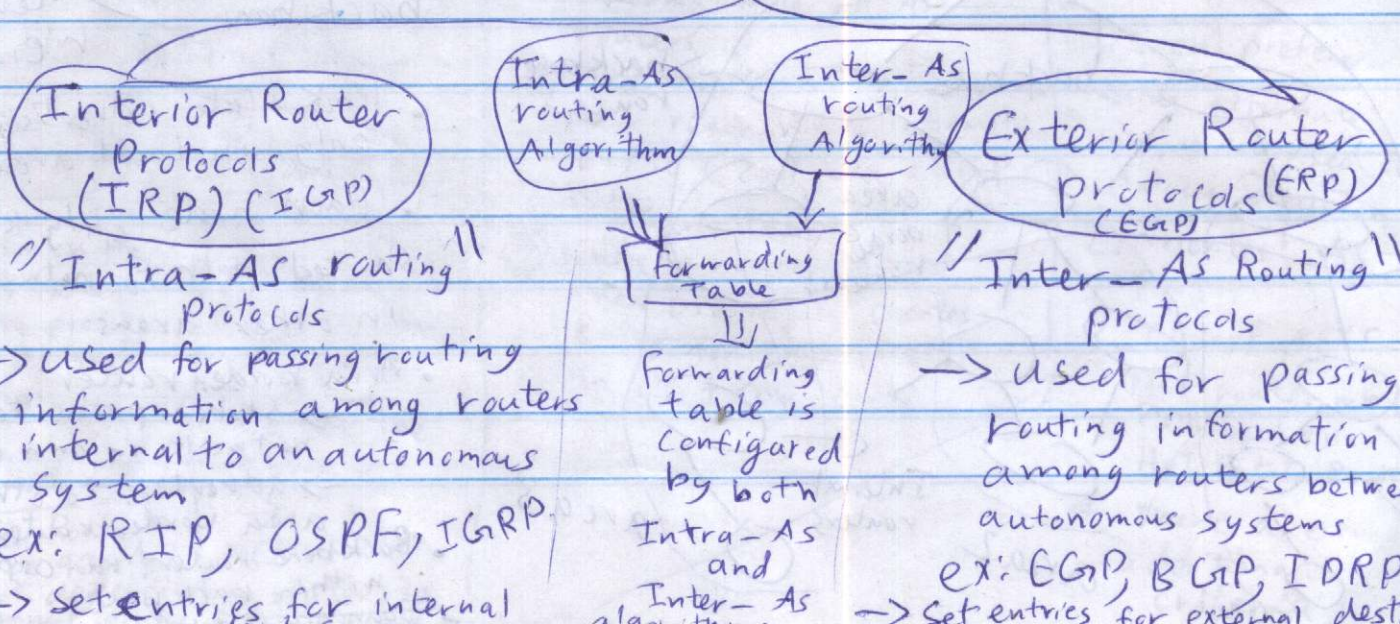
- link state advertisements only in local areas
- Each node only know shortest path to networks in other areas
- Area border router:
  → stores paths to networks in own area
  → advertises to other area border routers
- Backbone router: run OSPF within backbone
- Boundary

* Hierarchical routing:

- Why:
  - Scale: The number of the routers becomes large so, the effort of computing, storing and communicating is being bigger. ex: with 600 million dest.s, we can't store all dest.s in routing tables and routing tables update and exchange would use the links alone leaving transferring of data which is the goal from the internet.
  - Adminstrative autonomy: Each organization needs to ran its network as it wishes while still being able to connect its network to other outside networks

- Autonomous Systems (AS): an internet connected by homogeneous routers under the adminstrative control of a single entity (ex: Operated by same ISP or belonging to the same company network.)

- So, we have another point of view of Routing Protocols



// Intra-AS routing protocols
-> used for passing routing information among routers internal to an autonomous system
ex: RIP, OSPF, IGRP
-> set entries for internal

Forwarding table is configured by both Intra-AS and Inter-AS

// Inter-AS Routing protocols
-> used for passing routing information among routers between autonomous systems
ex: EGP, BGP, IDRP
-> set entries for external dest.s

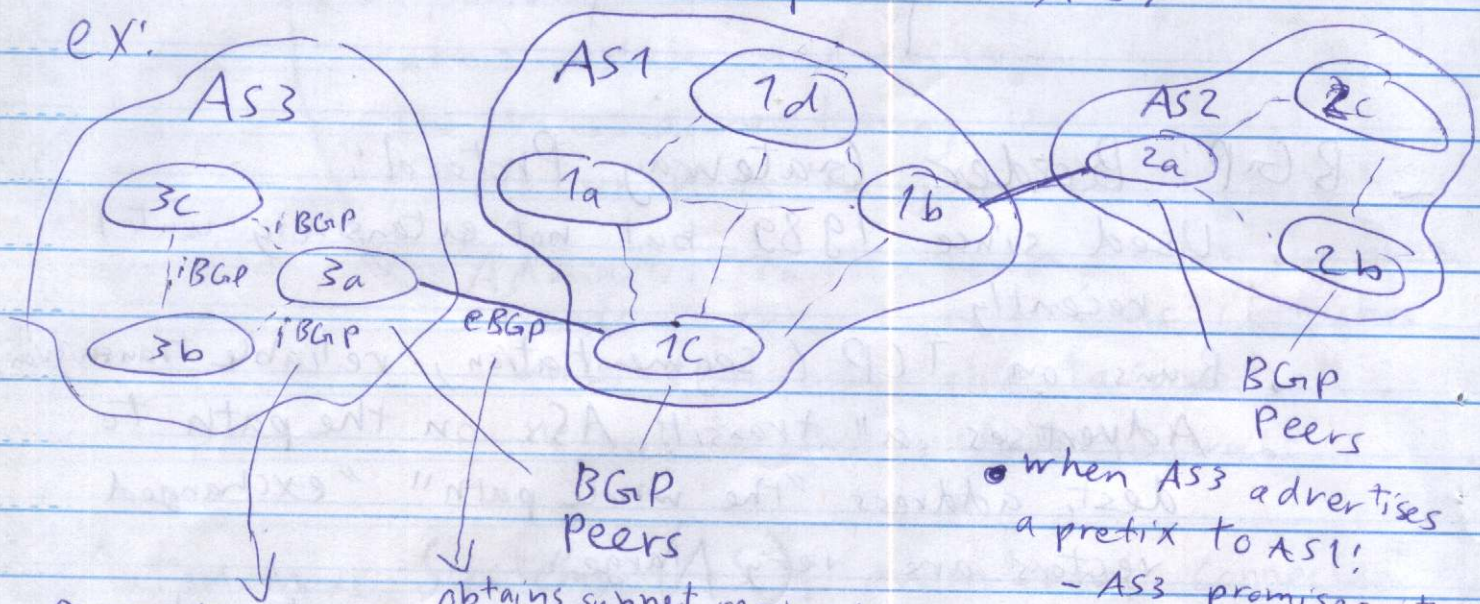
So, ⇒

	Distance Vector	Link State	
Intra-AS	RIP	OSPF	
Inter-AS			BGP

- BGP: Border Gateway Protocol:
  - Used since 1989 but not extensively until recently
  - Runs on TCP (segmentation, reliable transmission)
  - Advertises all transit ASs on the path to dest. address "the whole path" "exchanged vectors are very large"
  - A router may recieve multiple paths to a dest, so, it can choose the best
  - BGP provides each AS a means to
    - obtain subnet reachability from neighboring ASs.
    - Propagate the reachability information to all routers internal to the AS
    - Determine "good" routes based on reachability information and on AS policy
    - Allows each subnet to advertise its existence to the rest of the internet and BGP makes sure that all ASs in the internet know about the new subnet and how to reach it

- In BGP, pairs of routers exchange routing information over semi-permanent TCP connection using port 179.
  - iBGP session: it is a BGP connection between two routers but internal of the AS
  - eBGP session: it is a BGP connection between two routers which span two ASs.

ex:



propagates reachability information to all AS-internal routers

obtains subnet reachability information from neighboring ASs

- when AS3 advertises a prefix to AS1:
  - AS3 promises it will forward datagrams towards that prefix and it can aggregate prefixes in its advertisement

- Path attributes and BGP Route:
  → a Route is a prefix + some attributes cont. both contained within an advertisement:

attributes

- AS-PATH: Contains the ASs which the advertisement for the prefix has passed through
  ex: AS2 AS1
  Note: Route may be accepted or not depending on import policy of the AS
- NEXT-HOP: It is the router interface that begins the AS-PATH.
  ex: in the above ex → NEXT-HOP is router: 3a

① 3a sends prefix reachability information to 1c via eBGP session
② 1c can use iBGP to distribute new prefix information to all AS1 routers
③ 1b can re-advertise new reachability information to AS2 (2a) via eBGP between them,
④ when a router learns a new prefix, it creates entry for this prefix in its table

- BGP route selection depends on:
  1. Local preference value attribute which is based on policy decision
  2. Shortest AS-PATH
  3. Closest NEXT-HOP router which is based on HOT Potato Routing
  4. There are additional criteria

* Virtual Circuit Networks (VC) vs. Datagram Networks:

In transport layer we could have connectionless and connection-oriented services between 2 processes (UDP or TCP). Also, Network layer can provide connectionless or connection-oriented services between 2 hosts (Datagram Networks) and (Virtual Circuit Networks).

Virtual Circuit

- Connection Service
- with handshaking
- Host to Host
  - → "ATM and Frame" Relay Networks.
  - → "Used in Telephony"
- VC consists of:
  1. Path (series of links, routers)
  2. VC numbers (number for each link)
  3. Entries in forwarding table in each router along the path

→ The VC numbers are stored in the packet headers and they are replaced at each router with new ones obtained from the forwarding table on each router.

Datagram

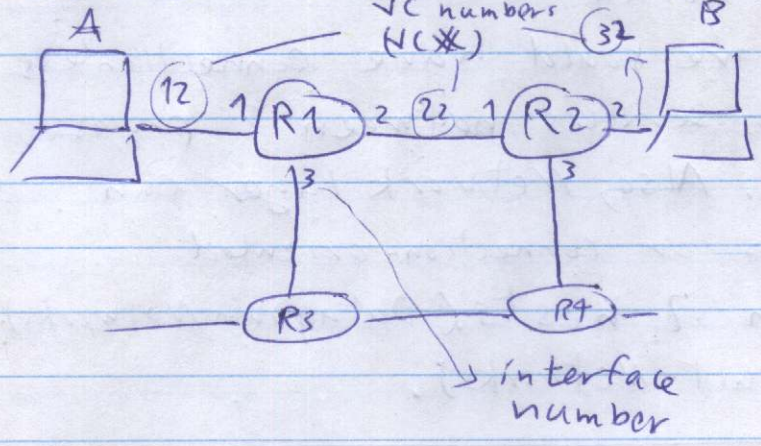
- Connectionless service
- No handshaking
- Host to Host

- when the src needs to send a packet, it stamps the packet with the dest. address and pops it to the network, there is no setup
- The routers then use the packet's dest. address to forward it using the forwarding table within each router to map dest. address to link interfaces

ex: ⇒ Forwarding table at a router:

Virtual Circuit

ex: we have this network:



Suppose that A requests to establish VC to B and the path is: A-R1-R2-B and assigns 12, 22, 32 as VC numbers,

→ The forwarding table of R1 could be something like that:

Incoming Interface	Incoming VC#	Outgoing interface	Outgoing VC#
1	12	2	22
2	63	1	18
3	7	2	17
1	97	3	87

→ when a VC is established an entry in the table is created and viceversa.

- VC Phases:
  - setup: Entries are added to the tables and path is determined.
  - Data Transfer: The packets flow from src. to dest.
  - VC Teardown: The src. inform the dest. that VC is terminated and the tables are updated. The src. inform through signaling messages (ms timescale) which use

Datagram

Dest. address range	Link interface
address 1 through address 2	0
address 3 through address 4	1
address 5 through address 6	2
otherwise	3

and ~~use~~ the router uses the longest prefix match.

- 1 to 5 minutes update.

Network Service Model: (Transport → Network):

- Services:

→ Guaranteed Delivery:

→ Guaranteed Delivery w/h bounded delay: delivery w/h specified (host to host) delay bound

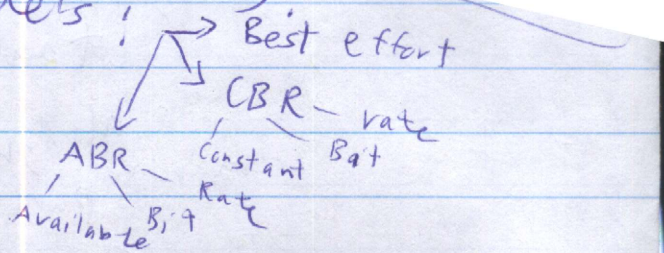
→ In-order packet delivery: packets in order

→ Guarenteed minimal Bandwidth: emulates behaviour of specified bit-rate link

→ Guaranteed maximum jitter: time bet. packet and packet in src = time bet packet and packet in dest.

→ Security services: Secret session

- Models:



Arch.	model	BW Guarantee	No-loss	Ordering	Timing	Congestion Ind.
Internet	Best Effort	None	None	Any order possible	Not maintained	None
ATM	CBR	✓ Const. Rate	✓	✓	✓	No Congestion
ATM	ABR	✓ minimum	X	✓	X	✓

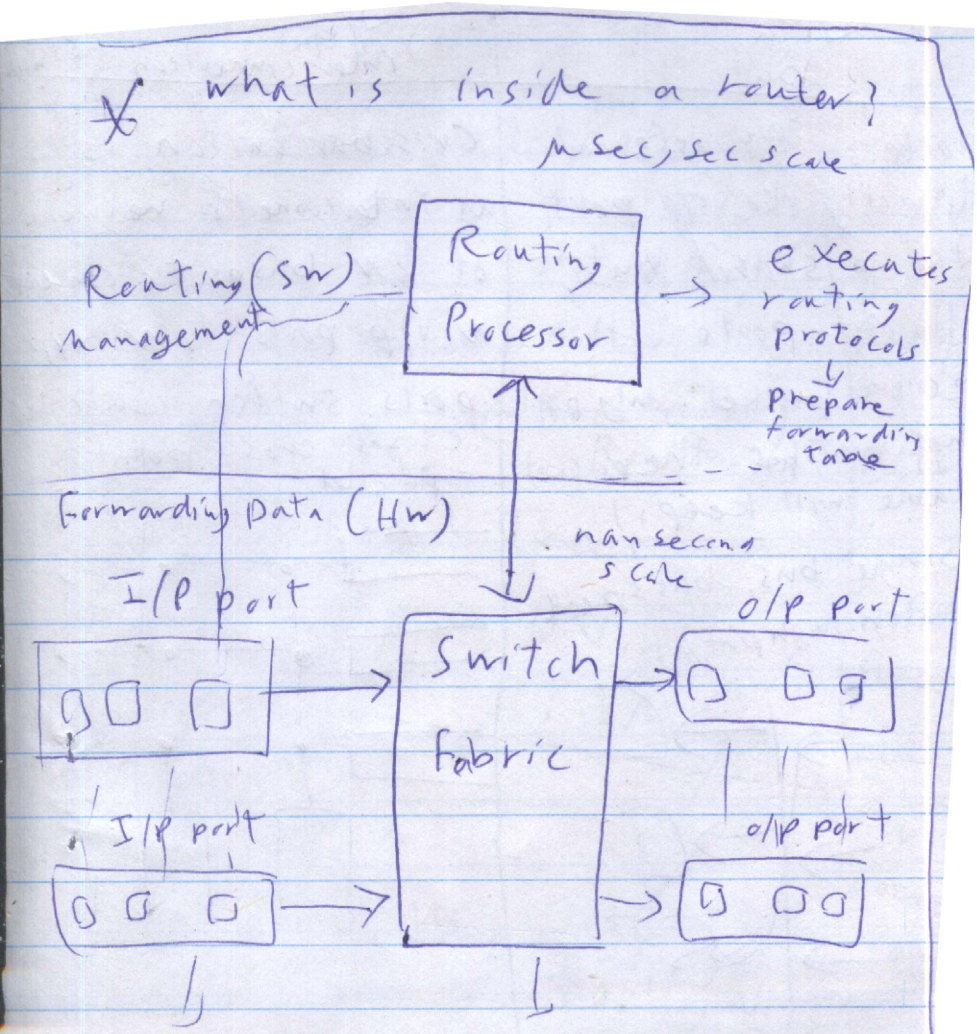
Const. bit rate audio and video traffic

slightly better than best effort

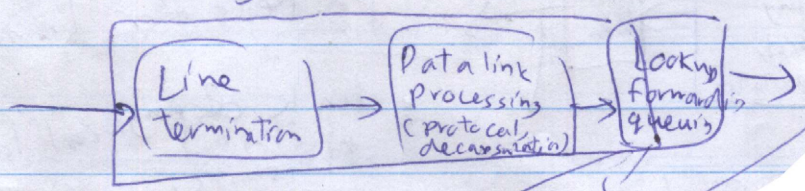
minimum Cell transmission Rate (MCR)

what is inside a router?

μsec, sec scale



- I/P processing:

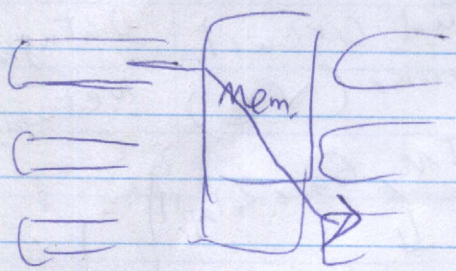


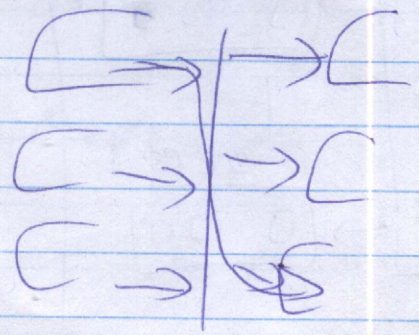
1. physical, link layer processing
2. Checksum, Packet life time
3. counters for network management

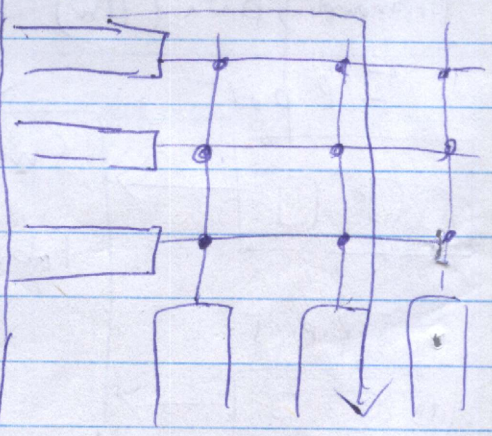
Blocked packets if packet uses the ~~fabric~~ switch fabric → Queue

- Switching Fabric:

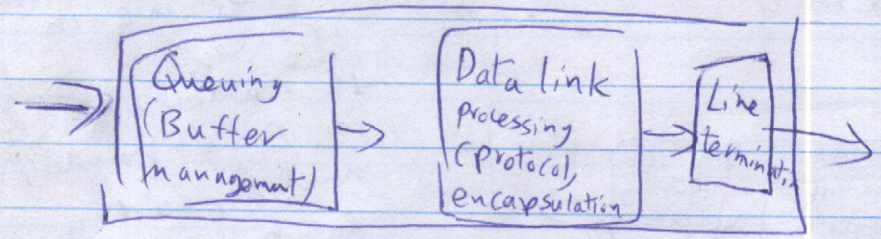
Switching via memory	via bus	via interconnection network
packet copied to processor memory → processor extracts dest. address → lookup the table → packet copied to o/p port ↓ memory BW will control throughput BW/2 ⌜ Read / write	packets transferred directly to o/p port via a shared bus, all o/p ports will recieve packet only port that matches the packet lable will keep it, Single bus, bus speed → throughput,	Crossbar switch is an interconnection network of 2N buses to connect N i/p ports to N o/p ports, switch controller closes cross point, parallel,







- output processing



- Selecting & dequeueing packets for transmission
- link & physical layer transmission fns.

- Routing control Plane:
  - centeralized calculations is better than distributed
  - Routing is separated in SW, HW